New issue

Security Vulnerability: Command Injection in multiple fast-filesystem-mcp tools due to unsafe use of child_process.execAsync #15

Open

123mutouren321414 opened last month

Summary

The MCP server ***fast-filesystem-mcp*** is vulnerable to command injection due to unsafe use of child_process.execAsync with user-controlled input in the ***fast_get_disk_usage***, ***fast_compress_files***, and ***fast_extract_archive*** tools.

Affected Versions

<= 3.5.1

Vulnerable Code

https://github.com/efforthye/fast-filesystem-mcp/blob/main/src/index.ts#L2211-L2240
https://github.com/efforthye/fast-filesystem-mcp/blob/main/src/index.ts#L550-L559
https://github.com/efforthye/fast-filesystem-mcp/blob/main/src/index.ts#L2215

Details

The MCP server fast-filesystem-mcp constructs command strings using user-supplied parameters and executes them via child_process.execAsync in multiple tools. Because execAsync invokes commands through a system shell, specially crafted input containing shell metacharacters (such as ;, &, or |) may be interpreted as additional commands rather than treated as data.
For example, an attacker could supply a malicious value in path to inject arbitrary shell commands, which would then be executed with the privileges of the MCP server process.
The vulnerability results from shell-based command execution combined with direct interpolation of untrusted input. In MCP environments, LLM-generated tool parameters influenced by external content may trigger execution of injected commands without direct local user interaction.

Impact

Successful exploitation allows attackers to execute arbitrary commands on the server hosting the MCP service. This may allow attackers to execute commands, access sensitive data, or modify the host environment depending on the privileges of the MCP server.

Recommendation

1. Don't use execSync. Use execFileSync instead, which pins the command and provides the arguments as array elements.
2. Apply strict input validation to all tool parameters exposed to MCP clients, especially path, output_path, archive_path, and extract_to parameters.
3. Use parameter separation with proper escaping to prevent shell command injection.

PoC

See the attached files:
fast-filesystem-mcp_bug.pdf

123mutouren321414 3 weeks ago Author •••

#16

123mutouren321414 2 weeks ago Author •••

security_advisory:
fast-filesystem-mcp_security_advisory.pdf

8endit 2 hours ago •••

Independently confirmed this via automated dynamic testing with mcpfuzz. Our scan found **17 of 25 tools** affected by unsanitized shell metacharacters in path parameters — significantly broader than the 3 tools identified here. Path traversal protection is solid (all 20 tests passed), but command injection surface is extensive. `compress_files` and `extract_archive` are highest risk due to likely `child_process` usage. Full scan report available on request.

Metadata

Assignees

No one assigned

Labels

No labels

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

Code with agent mode

No branches or pull requests

Participants